Omnimark vs XSLT

Presented by: David May

Omnimark vs XSLT

- Define each
- Similarities
- Differences
- Which one to use?
- Their roles today & future
- Optimal solution requirements
- Technical notes

Omnimark

A programming language -

- Powerful text processing language
- Built-in **SGML/XML** support
- Enterprise server support
- Designed for processing text in both static and server applications.
- Intended to read and write data to and from multiple sources.
- Process SGML, XML, HTML, and others
- Capable of transforming, splitting/merging, data to multiple formats.

XSLT

A stylesheet language –

- Extensible stylesheet language for transformations (XSLT) intended to provide the mechanism to transform and manipulate **XML** data to other formats.
- Commonly used to render XML (e.g. an XML to HTML conversion).

Omni/XSLT Similarities

- Both are capable of processing XML to other formats (e.g. HTML)

Omni/XSLT Differences

Process/ Function	Omnimark	XSLT
SGML	✓	
Any Text	✓	
XML	✓	✓
HTML	Input & Output	Output
File Input/Output	✓	
Data Handling	Very Large	Very Small
Variable support (a=b+1)	✓	
Multiple views of same content	Through processing to different outputs	✓
Type of Processing	Pre-Delivery	Post-Delivery
System Call support	✓	

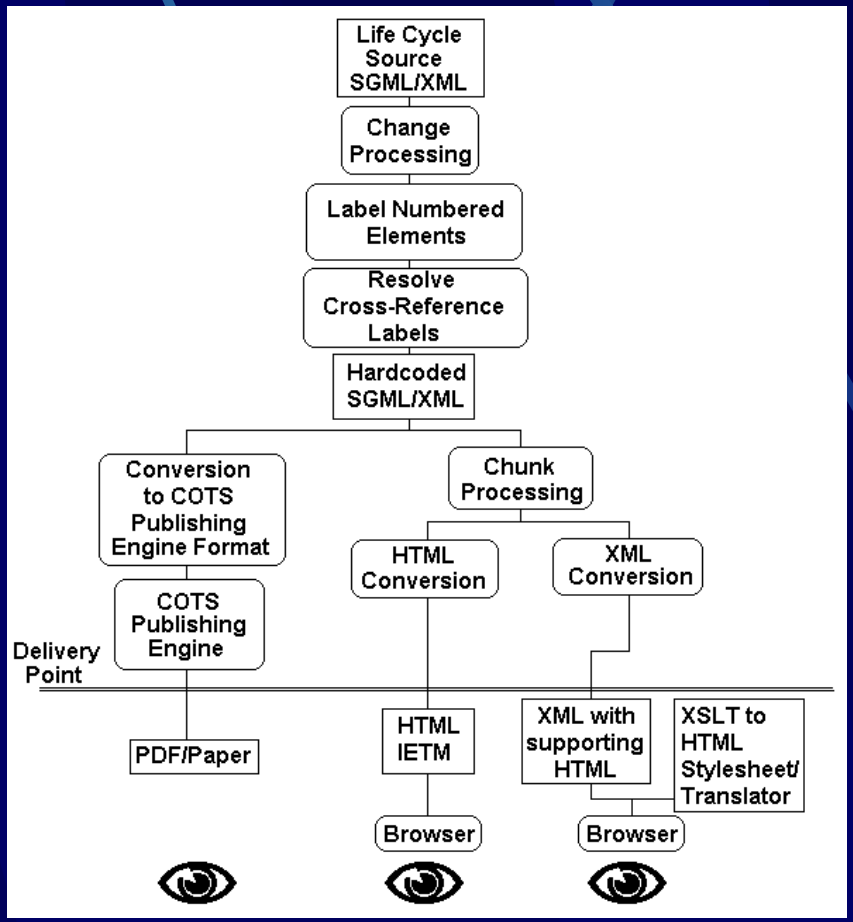
Which one to use?

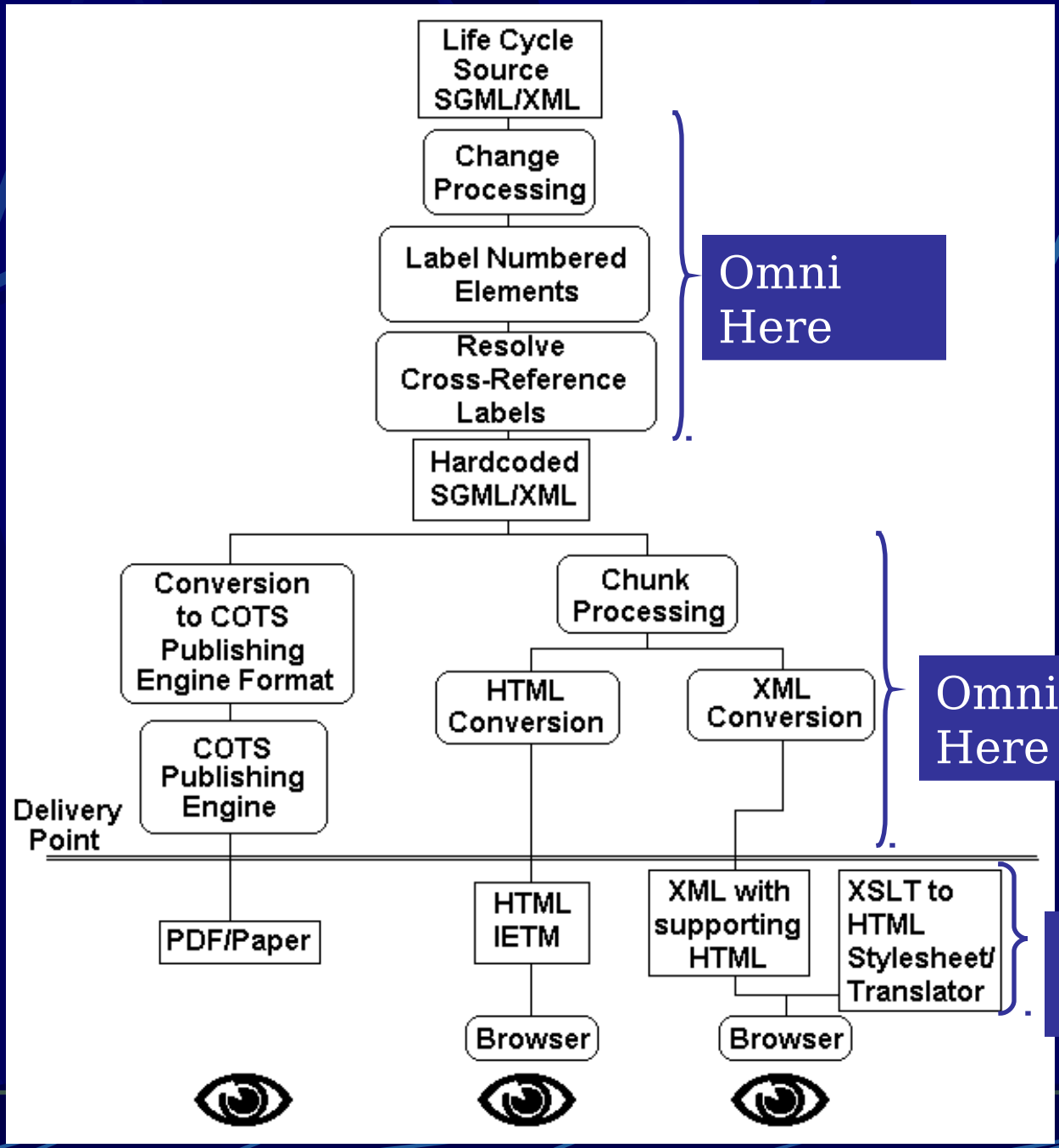
- This is not a good question!
- Omnimark is a programming language – XSLT is stylesheet language.
- While XSLT provides the ability to add styles to XML tags it was not originally intended to process data like a programming language.

A Better Question –
Which tool - when?

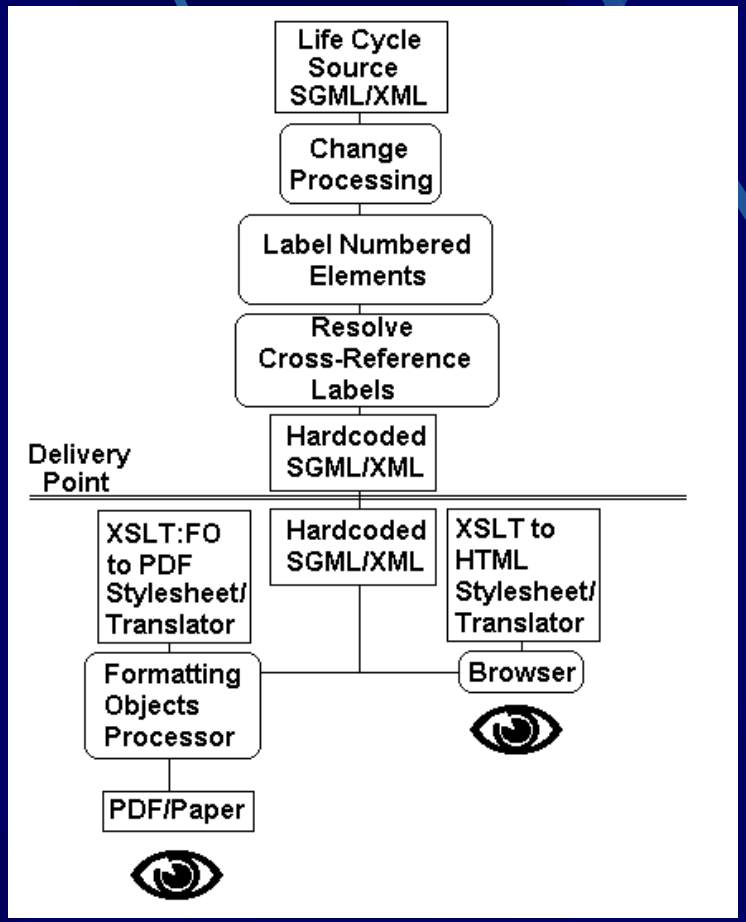
Omnimark	XSLT
Processing text	Working with small XML instances.
Creating files	Applying multiple views to the same content.
Using variables	
Adding labels to objects	
Breaking data apart or merging it together.	

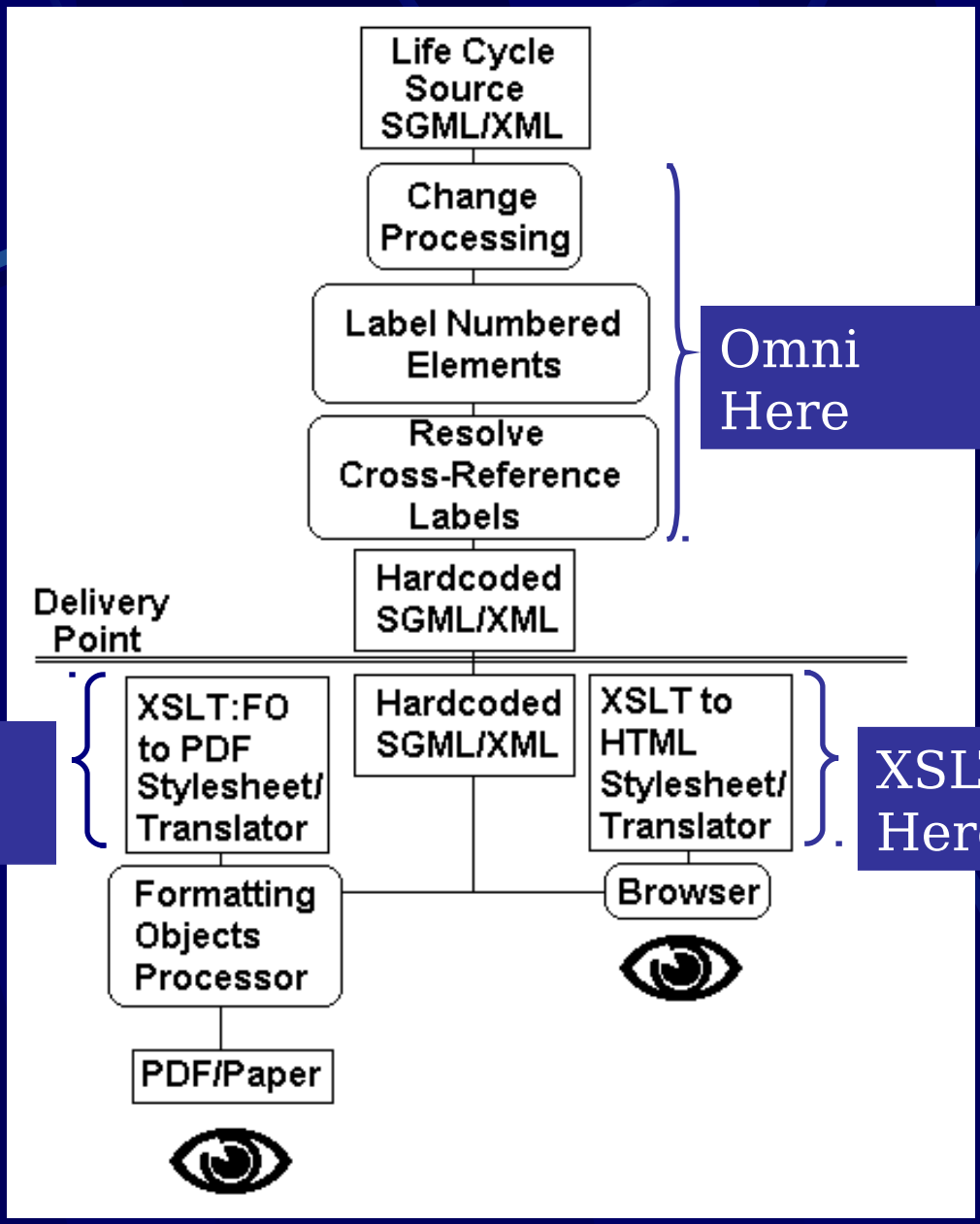
A Current Publishing System




Life Cycle Source SGML/XML
Change Processing
Label Numbered Elements
Resolve Cross-Reference Labels
Hardcoded SGML/XML
Omni Here
Conversion to COTS Publishing Engine Format
COTS Publishing Engine
Chunk Processing
HTML Conversion
XML Conversion
Omni Here
Delivery Point
PDF/Paper
HTML IETM
Browser
XML with supporting HTML
Browser
XSLT to HTML Stylesheet/ Translator
XSLT Here

A Future Publishing System




Life Cycle Source SGML/XML
Change Processing
Label Numbered Elements
Resolve Cross-Reference Labels
Omni Here
Hardcoded SGML/XML
Delivery Point
XSLT:FO to PDF Stylesheet/ Translator
Hardcoded SGML/XML
XSLT to HTML Stylesheet/ Translator
XSLT Here
XSLT Here
Formatting Objects Processor
Browser
PDF/Paper

To get there

- Changes are needed to:
  - XSLT technologies,
  - TM configuration management,
  - Deployment issues, and
  - Security.

To get there: (continued)

Needed XSLT technology changes

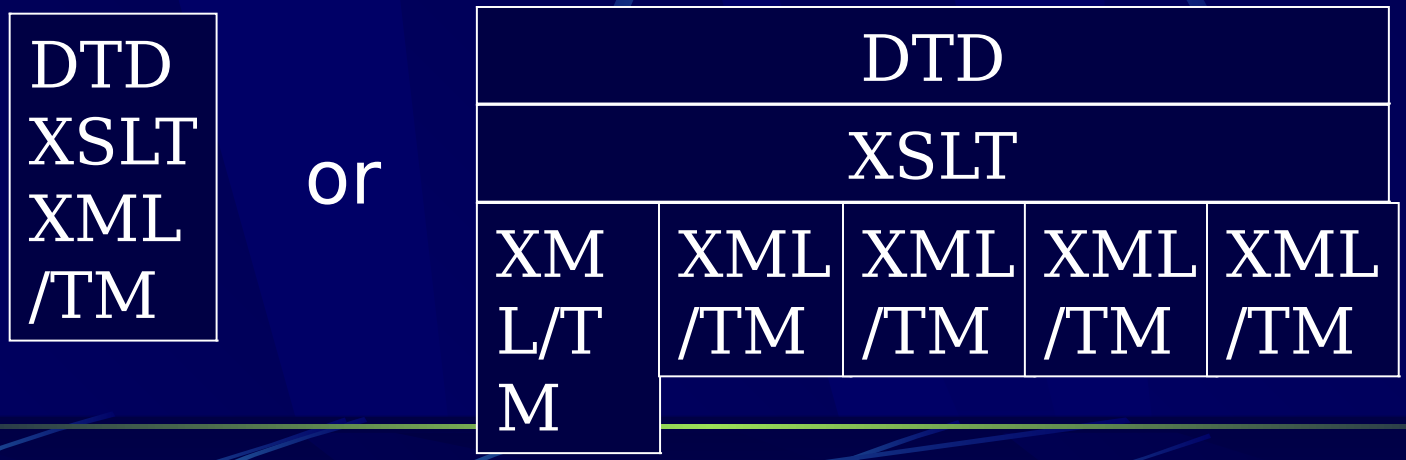
- Ability to quickly handle larger XML instances.
- Ability to use variables and rudimentary math functions in XSLT.
- Standard (built-in?) rendering engines need to emerge.

To get there: (continued)

Needed changes to configuration management?

With the splitting of content and formatting at delivery –

- What now constitutes a configuration unit?
- Where will responsibility for test/QC of configuration units and their interaction reside?



To get there: (continued)

Needed changes to configuration management?

- Will all sites:
  - Be using the same XSLT processor (e.g. Internet Explorer 6.0 with MSXML, MSXML2 v2.6, v3.0, or v4.0)?
  - Get the same results from those processors?

To get there: (continued)

Deployment Issues

- Translation processing will burden deployment platforms (anticipate need for new faster computers).
- Processing large XML instances on deployment platforms will increase RAM requirements and possibly hard drive space.

To get there: (continued)

Security

- Most current security software (e.g. firewalls) cannot process XML content and do not protect against what could be contained in that content.
- XSLT could translate apparently benign XML into damaging content such as a virus.

Technical Issues

- Converting XML to HTML puts additional processing burden on end-user computer.
- Current HTML features (e.g. search engines and TOCs) may need to be modified to work with XML.

Technical Issues

- When rendering large XML instances IE 6.0 will provide the “blue-screen of death”.
  - Creates need for “breaking up” manuals
    - Causes problems! For example, in XML a <volume> cannot start in one file and end in another!
- XML is evolving and XML support still has some issues (i.e. bugs).

Omnimark vs XSLT

Presented by: David May

To get there: (continued)

Security

- Most security software (e.g. firewalls) cannot process XML content and do not protect against what could be contained in that content Internet Week Article
- Could XSLT translate apparently benign XML into a damaging content such as a virus?